Shi, Y, Yu, DL, Tian, Y and Shi, Y

Modified Volterra model-based non-linear model predictive control of IC engines with real-time simulations

Modified Volterra Model Based Nonlinear Model Predictive Control of IC Engines with Real Time Simulations

Yiran Shi[1], Ding-Li Yu[2*], Yantao Tian[1] and Yaowu Shi[1]

[1] College of Communication Engineering, Jilin University, Changchun, China
[2] School of Engineering, Liverpool John Moores University, Liverpool, U.K.
* Corresponding author, Email: d.yu@ljmu.ac.uk

Abstract

Modelling of nonlinear dynamics of air manifold and fuel injection in an internal combustion (IC) engine is investigated in this paper using the Volterra series model. The Volterra model-based nonlinear model predictive control (NMPC) is then developed to regulate the air/fuel ratio at the stoichiometric value. Due to the significant difference between the time constants of the air manifold dynamics and fuel injection dynamics, traditional Volterra model is unable to achieve a proper compromise between model accuracy and complexity. A novel method is therefore developed in this paper by using different sampling periods, to significantly reduce the input terms while maintain the accuracy of the model. The developed NMPC system is applied to a widely used IC engine benchmark, the mean value engine model. Performance of the controlled engine under real-time simulation in the environment of dSPACE was evaluated. The simulation results show a significant improvement of the controlled performance compared with a feed-forward plus PI feedback control.

Keywords

Air-fuel ratio control, IC engines, Volterra model, nonlinear model predictive control, real-time simulations.

1 Introduction

In the past decades, the air/fuel ratio (AFR) control of internal combustion (IC) engines has

attracted more interests (Balluchi et al., 2000; Nicolao et al., 1996). To meet the new environmental requirements of government legislation, car manufacturers have been seeking ways to reduce emissions and fuel consumption while maintaining the engine performance. To develop effective control strategy, nonlinearity and interactions between variables such as engine speed, engine torque, spark ignition timing, fuel injection timing, air intake, AFR and others have been considered (Butt, *et al*. 2013; Tan and Mehrdad, 2000; Vinsonneau et al., 2003). AFR is regarded as one of the most important engine variables because of its relationship with fuel efficiency, emission reduction and driveability (Manzie et al., 2001; Manzie et al., 2002). The best balance between power output and fuel consumption can be obtained by maintaining AFR at the stoichiometric value of 14.7 with the three-way catalyst. Meanwhile, AFR is vital to emission control because of its ability to ensure the maximum efficiency of three-way catalyst (TWC) (Manzie et al., 2001; Manzie et al., 2002). For example, 1% change of AFR from the stoichiometric value will cause a significant increase in carbon monoxide (*CO*) and hydrocarbon (*HC*), and cause 50% NO_x increment (Manzie et al., 2001; Manzie et al., 2002).

In commercial electronic control unit (ECU), the AFR is controlled with a look-up table as feed-forward controller and compensated by a PI feedback control. There are some advantages in this method. However, it cannot produce desirable accurate control performance for highly nonlinear spark ignition (SI) engines ((Manzie et al., 2001; Manzie et al., 2002; Choi and Hendrick, 1998). Several alternative methods have been proposed to tackle the problem. For example, the sliding mode control developed by Shah, et al. (2015). Choi and Hendrick (1998) proposed an observer-based fuel injection control algorithm with sliding mode control to regulate the AFR. This method improved response speed and chattering of the AFR caused by sliding mode control, but the change of fuel film dynamics caused by aging or different fuel properties was not considered. Yoon and Sunwoo (2001) proposed an adaptive dynamic sliding-mode control to deal with the problems caused by engine uncertainties, but time-varying dynamics were not tackled.

In recent years with the increased computing speed, the model predictive control techniques have been attempted in internal combustion engines. Manzie et al. (2001) developed a radial basis function (RBF) neural network model for fuel injection dynamics. They found that this model was able to predict future air mass flow into the cylinder. Wang et al. (2006a) further developed an adaptive RBF model for AFR and the model-based NMPC for AFR control. In a followed paper Wang *et al*. (2006b) used the MLP neural network to model the IC engine and achieved similar performance. However, the neural network model is a grey box, in which little information can be used for analysis. The training of a MLP network model is quite time-consuming and may be trapped in a local minimum with the back-propagation training algorithm. Moreover, neural network model may take considerable computing time for output prediction, and consequently cannot be used in the NMPC algorithm for plants with fast dynamics in real applications.

In comparison, Volterra model has a simpler structure and is easier to identify. Therefore, it has been widely adopted for dynamic system modelling. This is the motivation for us to adopt the Volterra model in the NMPC for AFR control of IC engines. In the literature, the Volterra model has been used to model plants with slow dynamics (Zhang *et al*., 2009). Bryon *et al*. (1996) used a second-order Volterra model in NMPC for a simulated multivariable polymerization reactor. Gruber and Oliva (2012) had extended the application of the method in (Bryon *et al*., 1996) to the greenhouse temperature control. In further, Gruber et al. (2011) extended the application of Maner's method to control a PEM fuel cell using different training data. A literature search revealed that no research has been reported on Volterra modelling for AFR of IC engines. The main reason is that in AFR dynamics, air flow dynamics are slow while the fuel injection is very fast. This will need a big number of terms in the input of Volterra model, which is nearly impossible for an engine with fast dynamics.

A second-order Volterra model is used to model air manifold and fuel injection dynamics of a IC engine in this paper. The novel method proposed in this paper is to modify the conventional Volterra model by introducing different sampling times. In this way the truncation order can be selected relatively low to achieve a trade-off between the model accuracy and the model complexity. Then, the modified Volterra model-based NMPC is developed for AFR regulation against the dramatically and frequently change of throttle angle in air manifold. This is the major contribution of the paper. The model is developed using the input/output data from a nonlinear benchmark of IC engines, the mean value engine model (MVEM) (Hendricks et al., 1996). The modified Volterra model is on-line updated, so that the time varying effects of the engine physical parameters caused by aging, mechanical wear, etc. would be compensated. The developed control system is evaluated by the real-time simulation using dSPACE in conjunction with Matlab/Simulink.

In the remaining of the paper, IC engine dynamics are described in Section 2. Section 3 presents in detail the proposed novel modification to the traditional Volterra model. The modified Volterra model based NMPC is then presented in Section 4. The implementation of the developed method and the real time simulation results are described in Section 5. Finally some conclusions are drawn in Section 6.

2 SI engine dynamics

A widely used benchmark for engine modelling and control, the well-known MVEM developed by Hendricks et al. (1996) is used in this work. The configuration of an expanded MVEM is shown in Fig. 1 (where the AFR block and time delay block are added to the original model of Hendricks by

the authors). The model is decomposed into three sub-models: the first one describes the air intake manifold dynamics including manifold pressure and temperature, the second one describes crankshaft speed dynamics, and the third one describes the fuel injection dynamics. There are two inputs in this simulation model, the throttle angle $V(t)$ and the injected fuel mass flow rate $\dot{m}_{fi}(t)$, and one output AFR. All the variables in this section are defined in the nomenclature at the end of the paper. The configuration of the expanded MVEM is displayed in Fig.1.

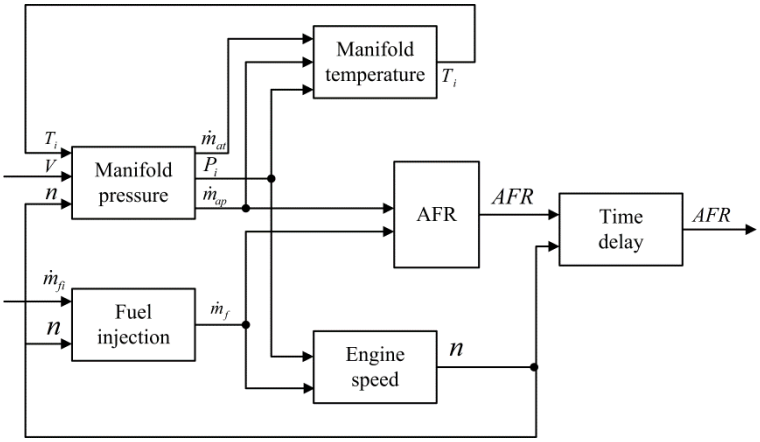

Fig. 1 The expanded MVEM in Simulink

2.1 Intake manifold filling dynamics

The intake manifold filling dynamics are analysed from the viewpoint of the air mass conservation inside the intake manifold. Two nonlinear differential equations are used to present the dynamics of manifold pressure $P_i(t)$ and manifold temperature, respectively. The manifold pressure $P_i(t)$ is a function of air mass flow rate past throttle plate $\dot{m}_{at}(t)$, the air mass flow into the intake port $\dot{m}_{ap}(t)$, the exhaust gas recirculation (EGR) mass flow rate $\dot{m}_{EGR}(t)$, the EGR temperature $T_{EGR}(t)$, the manifold temperature $T_i(t)$, and ambient temperature $T_a(t)$ as shown in (1).

$$\dot{P}_i = \frac{\kappa R}{V_i}(-\dot{m}_{ap}T_i + \dot{m}_{at}T_a + \dot{m}_{EGR}T_{EGR}) \tag{1}$$

The manifold temperature $T_i(t)$ dynamics are presented in (2).

$$\dot{T}_i = \frac{RT_i}{P_iV_i}[-\dot{m}_{ap}(\kappa-1)T_i + \dot{m}_{at}(\kappa T_a - T_i) + \dot{m}_{EGR}(\kappa T_{EGR} - T_i)] \tag{2}$$

Here EGR mass flow rate is set to zero because it is not considered in this simulation. In

equations (1) and (2), the air mass flow dynamics in the intake manifold can be described as follows. The air mass flow past throttle plate $\dot{m}_{at}$ is related with the throttle position and manifold pressure. The air mass flow into the intake port $\dot{m}_{ap}$ is represented by a speed–density equation as below:

$$\dot{m}_{at}(v, P_i) = m_{at1} \frac{P_a}{\sqrt{T_a}} \beta_1(v)\beta_2(P_r) + m_{at0} \tag{3}$$

$$\dot{m}_{ap}(n, P_i) = \frac{V_d}{120RT_i}(\eta_i \cdot P_i)n \tag{4}$$

where

$$\beta_1(v) = 1 - \cos(v) - \frac{v_0^2}{2!} \tag{5}$$

$$P_r = \frac{P_i}{P_a} \tag{6}$$

$$\beta_2(P_r) = \begin{cases} \sqrt{1 - \left(\frac{P_r - P_c}{1 - P_c}\right)^2}, & \text{if } P_r \geq P_c \\ 1, & \text{if } P_r < P_c \end{cases} \tag{7}$$

and $m_{at0}, m_{at1}, v_0, P_c$ are constants. In addition, it is easier to generate $\eta_i * P_i$, the quantity of normalised air charge, than to model the volumetric efficiency η_i directly. The normalised air charge can be obtained from the test of engine in the steady state and it is approximated with polynomial equation (8).

$$\eta_i \cdot P_i = s_i(n)P_i + y_i(n) \tag{8}$$

where $s_i(n)$ and $y_i(n)$ are positive, weak functions of the crankshaft speed, *n*, and $y_i \ll s_i$.

2.2 Crankshaft speed dynamics

The crankshaft speed is derived based on the conservation of rotational energy on the crankshaft.

$$\dot{n} = -\frac{1}{In}\left(P_f(P_i, n) + P_p(P_i, n) + P_b(n)\right) + \frac{1}{In} H_u \eta_i(P_i, n, \lambda)\dot{m}_f(t - \Delta\tau_d) \tag{9}$$

The friction power P_f and pumping power P_p are related to the manifold pressure P_i and crankshaft speed n. The load power P_b is a function of the crankshaft speed n. The volumetric efficiency η_i is a function of the manifold pressure P_i, crankshaft speed n and air fuel ratio λ.

2.3 Fuel injection dynamics

In Hendricks' model (Hendricks et al., 1996), the engine port fuel mass flow $\dot{m}_f$ is stated as,

$$\dot{m}_f = \frac{\dot{m}_{ap}}{L_{th}} \tag{10}$$

This means that the simulation model works in an ideal condition, where the AFR value is always equal to its stoichiometric value. Instead of using this ideal simulation of the injection system, a more practical fuel flow dynamic sub-model is considered (Hendricks, 2000).

$$\ddot{m}_{ff} = \frac{1}{\tau_f}(-\ddot{m}_{ff} + X_f \dot{m}_{fi}) \tag{11}$$

$$\dot{m}_{fv} = (1 - X_f)\dot{m}_{fi} \tag{12}$$

$$\dot{m}_f = \dot{m}_{fv} + \dot{m}_{ff} \tag{13}$$

This model represents the fuel flow dynamics of manifold injection engine by considering the fuel evaporation occurs in the intake manifold. The parameters in the model are the time constant for fuel evaporation τ_f and the proportion of the fuel which is deposited on the intake manifold or close to the intake valves X_f. These two parameters are operating point dependent and can be expressed in terms of the states of the model as

$$\tau_f(P_i, n) = 1.35 \times (-0.672n + 1.68) \times (P_i - 0.825)^2 + (-0.06 \times n + 0.15) + 0.56 \tag{14}$$

$$X_f(P_i, n) = -0.277 P_i - 0.055n + 0.68 \tag{15}$$

2.4 Air-fuel ratio measurement

In this simulation model, the AFR is calculated by using equation (16) as below. The air mass flow

into intake port $\dot{m}_{ap}$ is the output of intake manifold sub-model. The engine port fuel mass flow $\dot{m}_f$ is the output of fuel injection sub-model.

$$\lambda = \frac{\dot{m}_{ap}}{\dot{m}_f} \tag{16}$$

There are three causes of time delay for injection systems:

1) the two engine cycle delays between the injection of fuel and the expulsion from the exhaust valves;
2) the propagation delay for the exhaust gases to reach the oxygen sensor;
3) the sensor output delay.

It is found that the engine speed has more influence on these three delays than the manifold pressure. Therefore the following equation is used to represent the delays of injection systems (Manzie et al., 2001; Manzie et al., 2002).

$$t_d = 0.045 + \frac{10\pi}{n} \tag{17}$$

3 Modified Volterra model and engine modelling

3.1 Volterra model

A single-input single-output (SISO) second-order Volterra model, with the truncation order N_t, is defined as follows (Doyle et al., 2001).

$$y(k) = h_0 + \sum_{i=1}^{N_t} a_u(i)u(k-i) + \sum_{i=1}^{N_t}\sum_{j=i}^{N_t} b_u(i,j)u(k-i)u(k-j) \tag{18}$$

Equation (18) corresponds to a linear convolution model with additive nonlinear terms. In this model, $y(k)$ and $u(k)$ represent the measured output and input of the system at the current sampling instant *k*. The offset is denoted by h_0 and the coefficients of the linear and nonlinear terms are $a(i)$ and $b(i,j)$, respectively.

Similarly, the i^{th} output of a second-order Volterra model for a *q* inputs *p* outputs system is given as,

$$y_i(k) = h_{i0} + \sum_{l=1}^{q}\sum_{j=1}^{N_t} a^i_{l,j} u_l(k-j) + \sum_{l=1}^{q}\sum_{j=1}^{N_t}\sum_{n=j}^{N_t} b^i_{l,j,l,n} u_l(k-j)u_l(k-n), \quad i = 1,\cdots, p \qquad (19)$$

where N_t denotes the truncation order. There are no cross terms in the right hand side of equation (19). The term parameters of the linear and nonlinear terms for u_l are denoted with $a^i_{l,j}$ and $b^i_{l,j,l,n}$, respectively. h_{i0} is a bias term that can be fitted using input-output data.

3.2 Different response speeds in engine dynamics

For the AFR model to be developed one input is the injected fuel and one measured disturbance is the throttle angle. The Volterra model is therefore built with the two inputs and a single output. It is noted that the dynamics from the fuel injection to the AFR has a significant speed difference from that of the throttle angle to the AFR. This has been tested and clarified by applying a step change to each of the two inputs to the MVEM model and recorded the AFR responses. The test results are shown in Fig. 2(a) and 2(b). In Fig. 2(a) the applied fuel injection has a step change at *t* = 10 sec from 0.0005 kg/sec to 0.003 kg/sec, and the throttle angle is kept constant at 40 degrees. Fig. 2(b) depicts the AFR response when the fuel injection is maintained at 0.0015 kg/sec and the throttle angle has a step change at t = 10 sec from 20 to 60 degrees. It is measured that the transition time t_s in Fig. 2(a) is 8 sec (The AFR response to the step change of the fuel injection is composed of two parts. One part is very fast response while the rest is very slow response), while in Fig. 2(b) the transition time t_s is 0.8 sec. Therefore, the sampling time was chosen to be $T_s = \tau/10 = t_s/4/10 = 0.2\,\text{sec}$ for fuel injection, and $T_s = \tau/10 = t_s/4/10 = 0.02\,\text{sec}$ for the throttle angle.

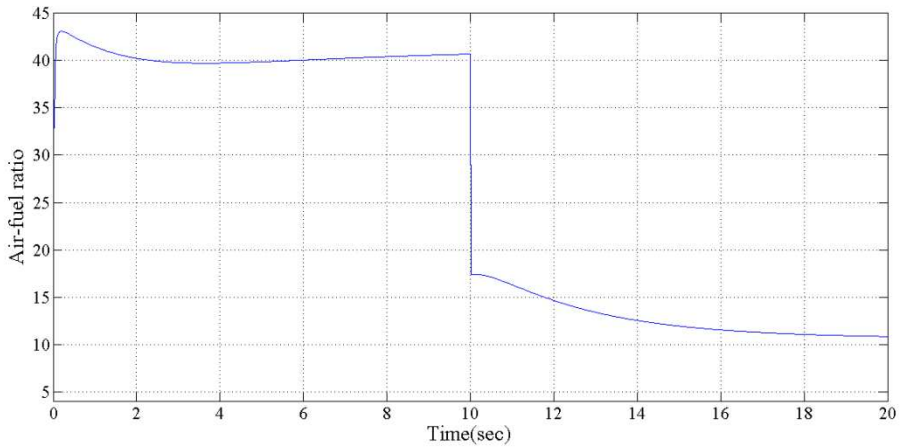

Figure 2(a) Air-fuel ratio response to a step change of fuel injection

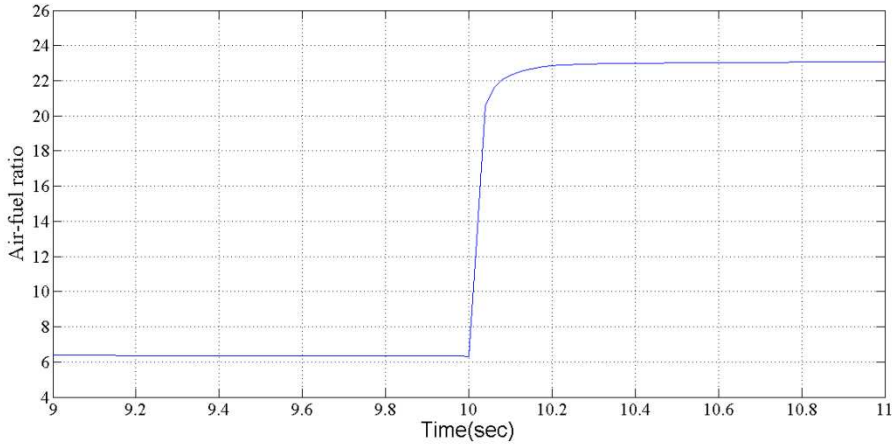

Figure 2(b) Air-fuel ratio response to a step change of throttle angle

The value of truncation order N_t has to be chosen carefully in relation to the dynamics of the engine, which is important for the modelling purpose. A small value of N_t will cause dynamic information in the data to be lost. On the other hand, if the value of N_t is too big then it will increase computing load dramatically. In this work, N_t is selected as 40. In order to avoid enormous computation, while to ensure the traditional Volterra model to cover every input dynamics, $T_s = 0.2\,\text{sec}$ is chosen. For throttle angle input, where the transition time is 0.8sec, only 4 data samples are generated for the whole sampling process and this causes significant loss of interested dynamic information. A modelling simulation of a traditional Volterra model with $T_s = 0.02\,\text{sec}$ and $N_t = 40$ is conducted, and the modelling results are shown in Fig. 3 in Section 3.6. It can be observed in Fig. 3 that the traditional second-order Volterra model is not appropriate to represent theICengine dynamics with the same sampling period for the two inputs. Thus, in this work, a modified Volterra model with different sampling periods is developed to model the dynamics with significant speed difference.

3.3 Modified Volterra model

As discussed in the previous section, the significant response speed difference between the two input variables causes difficulty in choosing sampling time and truncation order. In order to improve the performance of Volterra model, different sampling periods have been selected for these two inputs. Here t_u denotes the transition time of the fuel injection and t_v denotes the transition time of the throttle angle. This can be implemented by modifying the Volterra model equation (19) to the following equation.

$$y(k) = h_0 + \sum_{i=0}^{N_t-1} a_u(i)u(k - i \times T_u - 1) + \sum_{i=0}^{N_t-1}\sum_{j=i}^{N_t-1} b_u(i, j)u(k - i \times T_u - 1)u(k - j \times T_u - 1) + \sum_{i=1}^{N_t} a_v(i)v(k - i) + \sum_{i=1}^{N_t}\sum_{j=i}^{N_t} b_v(i, j)v(k - i)v(k - j) \quad (20)$$

where $T_u = \frac{t_u}{t_v} = 10$.

Firstly, the sampling time is chosen in this work as 0.01sec to ensure the high precision of the modified Volterra model. Besides, different sampling periods for the respective fuel injection is selected due to its longer transition time. For example at sample instant *k*, the samples $v(k-1), v(k-2), v(k-3), \ldots$ of throttle angle are selected according to equation (20), while the samples $u(k - i \times T_u - 1)$ of fuel injection are selected. As such, the fuel injection samples $u(k-1), u(k-11), u(k-21), \ldots$ will be used at sample instant k, and samples $u(k), u(k-10), u(k-20), \ldots$ will be used at sample instant $k+1$, until all samples have been selected. This modification enables the Volterra model to be implemented in short truncation order, while not losing necessary dynamic information.

3.4 Data collection

To excite the system covering all frequency spectrum and amplitude distributions of the nonlinear dynamics, the random amplitude sequences (RAS), the variable amplitude pseudo-random binary sequence (PRBS), the variable amplitude M-sequence signal, and Gaussian white noise were all used as the excitation signals to the engine model to acquire their own input/output data sets. The two engine inputs were the injected fuel and the throttle angle, while the one output was the air/fuel ratio. Each data set was used to identify a Volterra model with the same truncation order. Then, the mean squared modelling errors (MSE) of the test data set for each model were compared. It was found that the RAS as excitation signal gave the minimal MSE. So, the RAS was chosen and used in this work. In the data acquisition, the lower and upper bounds of the throttle angle was set as 20 and 60 degrees, while that for the injected fuel mass flow rate was set as 0.0005 and 0.003 kg/sec, which sufficiently cover the input operating space. The resulted air/fuel ratio output signal also covered the output operating space. The sampling period, *T*s = 0.02 sec, was used, which was considered appropriate for the engine dynamics. In this work, 20,000 input/output data samples were acquired.

3.5 Parameter estimation and model updating

The data set obtained has been normalised into a bounded range of (0, 1) using the linear scale (21)

before the data is used for identification. Then, the control variable optimised in the control algorithm is scaled back to normal data using (22).

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{21}$$

$$x = x_{scaled} * (x_{max} - x_{min}) + x_{min} \tag{22}$$

where x, x_{min}, x_{max} are data, minimal and maximal values within the data set respectively. In model identification, the parameters h_0, a_u, b_u, a_v, b_v in model (20) need to be estimated. The recursive Least Squares (RLS) algorithm was used to estimate these parameters. Firstly, the parameters are collected to form a parameter vector $\hat{\mathbf{w}} = \begin{bmatrix} h_0 & \mathbf{a}_u^T & \mathbf{b}_u^T & \mathbf{a}_v^T & \mathbf{b}_v^T \end{bmatrix}^T \in \Re^{N_t^2+3\times N_t+1}$. Then, the output $y(t)$ is presented as

$$y(t) = \hat{\mathbf{w}}^T \Psi(t) = \begin{bmatrix} h_0 & \mathbf{a}_u^T & \mathbf{b}_u^T & \mathbf{a}_v^T & \mathbf{b}_v^T \end{bmatrix} \bullet \begin{bmatrix} 1 \\ u(k) \\ [uu(k)] \\ v(k) \\ [vv(k)] \end{bmatrix} \tag{23}$$

where

$$uu(t) = \begin{bmatrix} (u(k-1) * u(k-1) \\ \vdots \\ u(k-1) * u(k-(N_t-1)T_u - 1) \\ \vdots \\ u(k-(N_t-1)T_u - 1) * u(k-1) \\ \vdots \\ u(k-(N_t-1)T_u - 1) * u(k-(N_t-1)T_u - 1) \end{bmatrix},$$

$$vv(t) = \begin{bmatrix} (v(k-1) * v(k-1) \\ \vdots \\ v(k-1) * v(k-(N_t-1)T_u - 1) \\ \vdots \\ v(k-(N_t-1)T_u - 1) * v(k-1) \\ \vdots \\ v(k-(N_t-1)T_u - 1) * v(k-(N_t-1)T_u - 1) \end{bmatrix}$$

As vector $\Psi(k)$ is a matrix of measurement data, parameter vector $\hat{\mathbf{w}}$ can be solved from (23) using the Least Squares algorithm. A RLS is used in this work for identification, which is not shown her as it can be found in a typical text book such as (Ljung, 1999).

3.6 Simulation results

As mentioned above, among the 20,000 collected data samples the first 15,000 samples are used for identification and the last 5,000 samples are used for model validation. The following initial values are used in the RLS algorithm.

$$\mathbf{P}(0) = 10^{8} \times \mathbf{I}_{n\times n},\ \ \hat{\mathbf{w}}(0) = 10^{-8} \times \mathbf{U}_{n\times 1},\ \ \lambda = 0.999$$

where $\mathbf{I}_{n\times n}$ is an identity matrix, $n = N_t^2 + 3\times N_t + 1$ and $\mathbf{U}_{n\times 1}$ is a vector with random entries. The modelling result of the modified Volterra model is shown in Fig. 4. For comparison, the same data set has also used to train a traditional Volterra model and the model prediction is displayed in Fig.3.

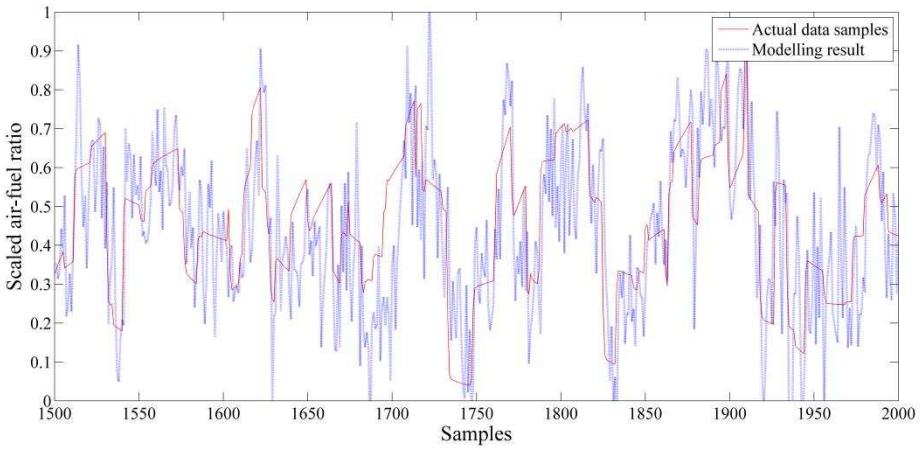

Figure 3 Test data and traditional Volterra model output

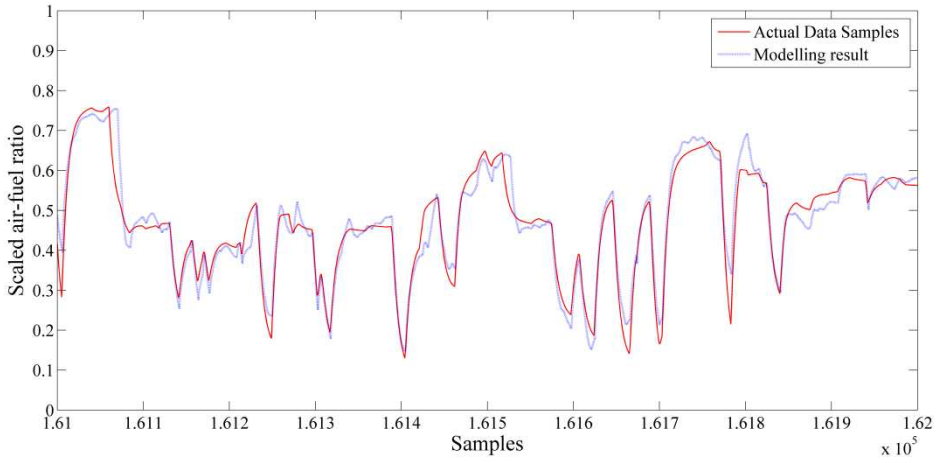

Figure 4 Test data and modified Volterra model output (MAE=0.0112)

It can be seen in Fig.3 and Fig.4 that the modified Volterra model outperformed the traditional

Volterra model.

4 NMPC based on modified Volterra model

4.1 System configuration

The strategy of NMPC for IC engines is shown in Fig. 5. Here the developed Volterra model is used to predict the future output, then the optimisation mechanism is used to determine the optimal control sequence, which minimise the objective function $J = \sum_{i=N_1}^{N_2}[r(i) - \hat{y}(i)]^2 + \eta\sum_{j=k}^{N_u}\Delta u(j)^2$. Finally the first element in the optimal control sequence is used to control the process, and the same procedure repeats at the next sampling period. The detailed procedure is given in the following sections.

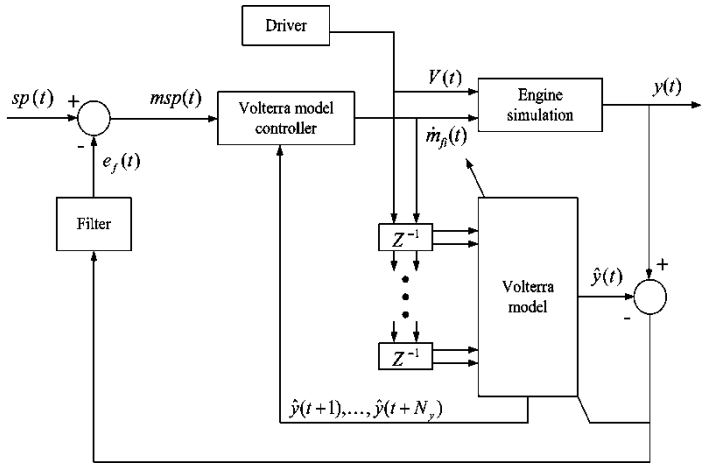

Figure 5 The adaptive modified Volterra model-based predictive control strategy

4.2 Model prediction

The modified Volterra model is used to predict the future behaviour of the IC engine considering the varying values in the inputs of throttle angle and fuel injection. This type of model can be considered as a logical extension of the models that are widely used in predictive controllers. At the control instant $k+1$, the future output of the identified nonlinear model (20) with the prediction horizon N_y, control horizon N_u, and truncation order N_t ($N_t \geq N_y \geq N_u$) can be calculated in a matrix form (Doyle et al., 2001) (in the rest of this Section, the reference to the traditional Volterra model based prediction control equations are all in (Doyle et al., 2001), if there is no special notice, the references will be omitted).

$$\mathbf{y}(k+1) = G_u \mathbf{u}(k) + \mathbf{f}_u(k+1) + \mathbf{c}(k+1) \tag{24}$$

Here the predicted output vector $\mathbf{y}(k+1) \in \Re^{N_y}$ and the future input sequence $\mathbf{u}(k) \in \Re^{N_t}$ are defined as follows,

$$\mathbf{y}(k+1) = \left[y(k+1),\, y(k+2), \ldots,\, y(k+N_y)\right]^T$$

$$\mathbf{u}(k) = \left[u(k), u(k+1), \ldots, u(k+N_u-1)\right]^T$$

And, $G_u \in \Re^{N_y \times N_u}$ is the linear part of the future fuel injection input, the vector $\mathbf{f}_u(k+1) \in \Re^{N_y}$ contains the future–future cross terms of the input sequence $\mathbf{u}(k)$. The term $\mathbf{c}(k+1) \in \Re^{N_y}$ contains only terms which do not depend on the current or future inputs and is defined as,

$$\mathbf{c}(k+1) = \mathbf{c}_u(k-1) + \mathbf{c}_v(k+1) + \mathbf{h}_0 + \mathbf{d}(k+1) \tag{25}$$

with

$$\mathbf{c}_u(k-1) = H_u \mathbf{u}_p(k-1) + \mathbf{g}_u(k+1) \tag{26}$$

and

$$\mathbf{c}_v(k+1) = H_v \mathbf{v}_p(k-1) + \mathbf{g}_v(k+1) + G_v \mathbf{v}(k) + \mathbf{f}_v(k+1) \tag{27}$$

where $\mathbf{c}_u(k-1) \in \Re^{N_y}$ represents the constant term depending on the vector of past first input values; $\mathbf{c}_v(k+1) \in \Re^{N_y}$ is the vector containing the terms of the future second input values. The vectors $\mathbf{h}_0 \in \Re^{N_y}$ and $\mathbf{d}(k+1) \in \Re^{N_y}$ include the offset of the identified model (20) and the estimation error in the prediction, and are defined as $\mathbf{h}_0 = [h_0 \;\; \cdots \;\; h_0]^T$ and $\mathbf{d}(k+1) = [d(k) \;\; \cdots \;\; d(k)]^T$, respectively. The terms $H_u \mathbf{u}_p(k-1)$ with $H_u \in \Re^{N_y \times N_t}$ and $\mathbf{g}_u(k-1) \in \Re^{N_y}$ are the linear and nonlinear parts of the past input values, respectively. Analogously, $H_v \mathbf{v}_p(k-1)$ with $H_v \in \Re^{N_y \times N_t}$ and $\mathbf{g}_v(k-1) \in \Re^{N_y}$ represents the linear and nonlinear terms of the past second input values. Furthermore, $G_v \mathbf{v}(k)$ with $G_v \in \Re^{N_y \times N_u}$ and $\mathbf{f}_v(k+1) \in \Re^{N_y}$ denote the linear and nonlinear terms of the future second input values. The past first input vector $\mathbf{u}_p(k-1) \in \Re^{N_t}$ and the past second input vector $\mathbf{v}_p(k-1) \in \Re^{N_t}$ are defined as

$$\mathbf{u}_p(k-1) = \left[u(k-1), u(k-2), \ldots, u(k-N_t)\right]^T$$

$$\mathbf{v}_p(k-1) = [\underbrace{v(k-1), v(k-1), \ldots, v(k-1)}_{T_u}, \underbrace{v(k-2), v(k-2), \ldots, v(k-2)}_{T_u},$$

$$\ldots, \underbrace{v(k-N_t \times t_u / t_v), v(k-N_t \times t_u / t_v), \ldots, v(k-N_t \times t_u / t_v)}_{T_u}]^T \quad (31)$$

The future values of the throttle angle input are assumed constant. The vector $\mathbf{v} \in \Re^{N_u}$, are defined by

$$\mathbf{v} = [v(k), v(k), \ldots, v(k)]^T \quad (32)$$

As a consequence, the linear and nonlinear terms $G_v\mathbf{v}$ and $\mathbf{f}_v$ that depend on the future values of the throttle angle input are calculated with the current values of the throttle angle input. In order to expand the explanation in this algorithm, the terms G_u, G_v, H_u, H_v, B_u, B_v, $\mathbf{f}_u(k+1)$, $\mathbf{f}_v(k+1)$, $\mathbf{g}_u(k+1)$, $\mathbf{g}_v(k+1)$ are described as follows. With $m = \{u, v\}$, the matrix G_m is given as,

$$G_m = \begin{bmatrix} a_m(1) & 0 & \cdots & 0 \\ a_m(2) & a_m(1) & \ddots & 0 \\ \vdots & \vdots & \ddots & a_m(1) \\ \vdots & \vdots & \ddots & a_m(1)+a_m(2) \\ \vdots & \vdots & \ddots & \vdots \\ a_m(N_y) & a_m(N_y-1) & \cdots & \sum_{i=1}^{N_y-N_u+1} a_m(i) \end{bmatrix} \quad (33)$$

Similarly, the matrix H_m with $m = \max\{u, v\}$ represents the linear term of the past inputs and are defined generally as,

$$H_m = \begin{bmatrix} a_m(2) & a_m(3) & \cdots & a_m(N_t-1) & a_m(N_t) & 0 \\ a_m(3) & a_m(4) & \cdots & a_m(N_t) & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_m(N_t-1) & a_m(N_t) & \cdots & 0 & 0 & 0 \\ a_m(N_t) & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix} \quad \text{if } N_y = N_t, \quad (34)$$

$$H_m = \begin{bmatrix} a_m(2) & a_m(3) & a_m(4) & a_m(5) & \cdots & a_m(N_t) & 0 \\ a_m(3) & a_m(4) & a_m(5) & \cdots & a_m(N_t) & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_m(N_y+1) & \cdots & a_m(N_t) & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{if } N_y < N_t. \quad (35)$$

The vector $\mathbf{f}_u(k+1) \in \Re^{N_y}$ contains the future-future and future-past cross terms of the fuel injection input, and is given in the following equations.

$$\begin{aligned} f_u(k+i) = &[u(k+i-1) \quad u(k+i-2) \quad \cdots \quad u(k) \quad 0 \quad \cdots \quad 0] \cdot B_u \\ &\cdot [u(k+i-1) \quad u(k+i-2) \quad \cdots \quad u(k) \quad u(k-1) \quad \cdots \quad u(k-N_t+3)]^T \end{aligned} \qquad i = 1, \cdots, N_y \tag{36}$$

In this project for the inputs with different varying speeds, the vector $\mathbf{f}_v(k+1) \in \Re^{N_y}$ needs to be changed as,

$$\begin{aligned} f_v(k+i) = &[v(k+i-1) \quad v(k+i-2) \quad \cdots \quad v(k) \quad 0 \quad \cdots \quad 0] \cdot B_v \\ &\cdot [\underbrace{v(k+i-2) \quad v(k+i-2)}_{i-1} \quad \underbrace{v(k) \quad \cdots \quad v(k)}_{T_u} \quad \underbrace{v(k-1) \quad \cdots \quad v(k-1)}_{T_u} \\ &\cdots \underbrace{v(k-N_t \times t_u / t_v) \quad \cdots \quad v(k-N_t \times t_u / t_v)}_{T_u}]^T \end{aligned} \qquad i = 1, \cdots, N_y \tag{37}$$

where $v(k) = v(k+1) = v(k+2) = \cdots$, and $\mathbf{g}_u(k+1) \in \Re^{N_y}$ contains the past-past cross terms and is defined as below,

$$\begin{aligned} g_u(k+3) = &[\underbrace{0 \quad \cdots}_{i} \quad 0 \quad u(k-1) \quad u(k-2) \quad \cdots \quad u(k-N_t+1)] \cdot B_u \cdot \\ &[\underbrace{0 \quad \cdots}_{i} \quad 0 \quad u(k-1) \quad u(k-2) \quad \cdots \quad u(k-N_t+1)]^T \end{aligned} \qquad i = 1, \cdots, N_y \tag{38}$$

In the same way as equation (35), $\mathbf{g}_v(k+1) \in \Re^{N_y}$ can be written as:

$$\begin{aligned} g_v(k+2) = &[\underbrace{0 \quad \cdots \quad 0}_{i} \quad \underbrace{v(k-1) \cdots v(k-1)}_{T_u} \quad \underbrace{v(k-2) \cdots v(k-2)}_{T_u} \\ &\cdots \underbrace{v(k-N_t \times t_u / t_v) \cdots v(k-N_t \times t_u / t_v)}_{T_u}] \cdot B_v \\ &\cdot [\underbrace{0 \quad \cdots \quad 0}_{i} \quad \underbrace{v(k-1) \cdots v(k-1)}_{T_u} \quad \underbrace{v(k-2) \cdots v(k-2)}_{T_u} \\ &\cdots \underbrace{v(k-N_t \times t_u / t_v) \cdots v(k-N_t \times t_u / t_v)}_{T_u}]^T \end{aligned} \qquad i = 1, \cdots, N_y \tag{39}$$

The matrix $B_m \in \Re^{N_t \times N_t}$ with $m = \{u, v\}$ used above is defined as,

$$B_m = \begin{bmatrix} b_m(1,1) & b_m(1,2) & b_m(1,3) & \cdots & b_m(1,N_t) \\ 0 & b_m(2,2) & b_m(2,3) & \cdots & b_m(2,N_t) \\ 0 & 0 & b_m(3,3) & \cdots & b_m(3,N_t) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & b_m(N_t,N_t) \end{bmatrix} \tag{40}$$

With the above definitions and detailed explanation, the future output prediction can be calculated following these equations.

4.3 Optimal control algorithm

The developed modified Volterra model is used as the internal model to predict future output of the engine for the model predictive control. The objective function for the optimization is set as,

$$\begin{cases} J = [\mathbf{s} - \mathbf{y}(k+1)]^T \boldsymbol{\Lambda}_1 [\mathbf{s} - \mathbf{y}(k+1)] + \mathbf{u}(k)^T \boldsymbol{\Lambda}_2 \mathbf{u}(k) \\ \text{s.t.} \quad u_{min} \le \mathbf{u}(k) \le u_{max} \end{cases} \tag{41}$$

where $\boldsymbol{\Lambda}_1 \in \Re^{N_y \times N_y}$ and $\boldsymbol{\Lambda}_2 \in \Re^{2N_u \times 2N_u}$ are the weight matrices that are positive definite, the u_{min}, u_{max} are the lower and upper bounds of the input imposed by the actuator. Substituting (24) into (41) we have,

$$\begin{aligned} J &= [s - G_u u(k) - f_u(k+1) - c(k+1)]^T \Lambda_1 [s - G_u u(k) - f_u(k+1) - c(k+1)] + \Delta u(k)^T \Lambda_2 \Delta u(k) \\ &= [r(k) - G_u u(k)]^T \Lambda_1 [r(k) - G_u u(k)] + \Delta u(k)^T \Lambda_2 \Delta u(k) \\ &= r^T(k) \Lambda_1 r(k) - 2r^T(k) \Lambda_1 G_u u(k) + u^T(k) [G_u^T \Lambda_1 G_u + \Lambda_2] u(k) \\ &= \text{cons tan t} + u^T(k) H u(k) + G u(k) \end{aligned} \tag{42}$$

where $r(k) = s - f_u(k+1) - c(k+1)$, $H = G_u^T \Lambda_1 G_u + \Lambda_2$ and $G = -2r^T(k) \Lambda_1 G_u$

The constraints are linear inequality,

$$Qu(t) \le p$$

where

$$Q = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad p = \begin{bmatrix} u_{max} \\ -u_{min} \end{bmatrix}$$

Now, we have a standard optimization problem of quadratic programing,

$$J(k) = \frac{1}{2} u^T(k) H u(k) + G u(k) \tag{43}$$

subject to

$$Qu(t) \leq p \tag{44}$$

For the quadratic programming two developed methods, the active set method and the interior point method have been typically used. It was reported that between the two the active set method is faster for small and medium size systems (Coen et al., 2008). Therefore, the active set method is used in this study to find out optimal control. As the method has been well developed it will not be described in detail here. Due to the two reasons: one is that only the inequality constraint will be applied in the optimization, and the other is that it is preferred to execute the optimal programme as fast as possible, the active set strategy is briefly presented to help understand the programing.

The active set method starts from an initial feasible point that is as close to the optimal solution as possible. At the same time an active set A_k (a set of active constraints, i.e. those for which the solution points are on the boundaries) is formed and updated in every iterative step. The necessary and sufficient conditions for u to be the global optimum for convex QP problem are given by the Karush-Kuhn-Tucker (KKT) condition (Fletcher, 1987). The active set method, after calculating the initial feasible point, will determine a series of feasible points that converge to the optimal solution. The search direction Δu is calculated to minimize the objective function while to be remained on the any active constraint boundaries. The feasible subspace for Δu is formed from a basis Z_k whose columns are orthogonal to the estimate of the active set A_k. Thus, Δu can be formed from the basis vectors of the null space of A_k.

A linear combination of the vectors of the null space Z_k is used as below

$$\Delta u(k) = Z_k q_k \tag{45}$$

where q_k is the coefficient vector. Then, we substitute the search direction (45) into objective function (43), and view the objective function as a function of coefficient vector p.

$$J(k) = \frac{1}{2} q_k^T Z_k^T H Z_k q_k + G Z_k q_k \tag{46}$$

Differentiating (46) with respect to q_k yields

$$q_k = -\frac{Z_k^T G}{Z_k^T H Z_k} \tag{47}$$

This leads to the search direction,

$$\Delta u(k) = -\frac{Z_k^T G}{Z_k^T H Z_k} * Z_k \tag{48}$$

Thus, the optimal control can be updated in each iteration as follows,

$$u(k+1) = u(k) + \alpha \Delta u(k) \tag{49}$$

where α is the step length. Due to the quadratic nature of the objective function, there are only two choices of α , A step of unity along Δu is the exact step to the minimum of the function restricted to the null space of A_k, If such a step can be taken without violation of the constraints, then this is the solution to the QP. Otherwise, the step along Δu to the nearest constraint is less than unity and a new constraint is included in the active set at the next iteration. The distance to the constraint boundary in the direction Δu is taken as the step length,

$$\alpha = \min\{ \frac{-[A_i u(k) - p_i]}{A_i \Delta u(k)} \}, \quad i = 1, \cdots, m \tag{50}$$

The global optimal solution can be checked if the KKT condition is satisfied.

5 Performance Evaluations

The performance of the developed model predictive control has been assessed using both computer simulation with Matlab/Simulink, and real time simulation using dSPACE facilities. The first evaluation is to test if the developed method is effective in regulating the AFR to the stoichiometric value when the engine is subjected to changes of throttle angle. Whilst the second evaluation is to test if the developed method can be practically executed in the required time period and achieve similar results as that of the first evaluation when the engine is under the same disturbance. Due to the fact that our test engine is not equipped with alternative control injection function, the real engine test cannot be implemented in this work and will be done in the future with the test engine in other institute.

In this paper, a real-timeICengine AFR control simulation platform based on dSPACE components, dSPACE MicroAutoBox and Matlab/Simulink is established. By using RTI software provided by dSPACE the MEVM Simulink model is downloaded to the DS1005PPC processor board to simulate the plant to be controlled. Download the control algorithm in this paper to the dSPACE MicroAutoBox (DS1401), and communicate with DS1005ppc processor board by RS232 to achieve the real-time AFR control. Simulation has been done in the real-time environment (as shown in Fig.6) with the sample period of 0.02s, by using ControlDesk software in the observer computer to display and record the dynamic signals.

Fig.6 Real-time simulation environment

As the major disturbance the throttle angle of a near step change from 25 degrees to 50 degrees with 0.5 degree uncertainty is used in the evaluation. The disturbance is shown in Fig.7. As mentioned before, the stoichiometric value of AFR must be controlled in the boundary of 14.7±1%. The sampling time is chosen as 0.02s. Through several times tuning, the prediction horizon N_y =36 and the control horizon N_u =15 are chosen. The weighting matrices Λ_1 and Λ_2 in the objective function (36) are chosen as two diagonal matrices with dimension of N_y and N_u, and the amplitude of 1 and 0.1, respectively. With the chosen parameters above, the NMPC control has been successful and satisfactory simulation results were obtained. As the Matlab/Simulink simulation results are similar to that of the real-time simulation, only the real-time simulation results are displayed here.

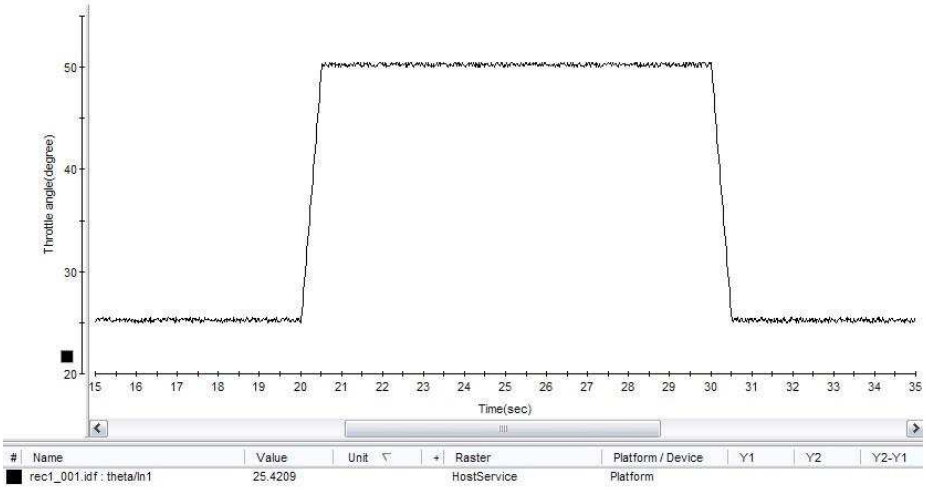

Figure 7 Throttle angle change during control

The regulated AFR is shown in Fig. 8(a) and the optimal control, the injected fuel flow rate is

shown in Fig.9. In Fig. 8(a) the required AFR range, the stoichiometric value of 14.7 with 1% error, $14.7(1\pm1\%)$ is drawn in red line. From the figures it is evident that the AFR achieved under the NMPC is well tuned and is within the boundary of $14.7(1\pm1\%)$. When the throttle angle has a step change, the AFR jumps out of the boundary a little bit and is tuned back into the boundary quickly. To have a clearer view for the dynamic tuning effects the AFR response at t=20 sec and t=30 sec corresponding to the step change of the throttle angle are augmented and shown in Fig. 8(b) and 8(c), respectively.

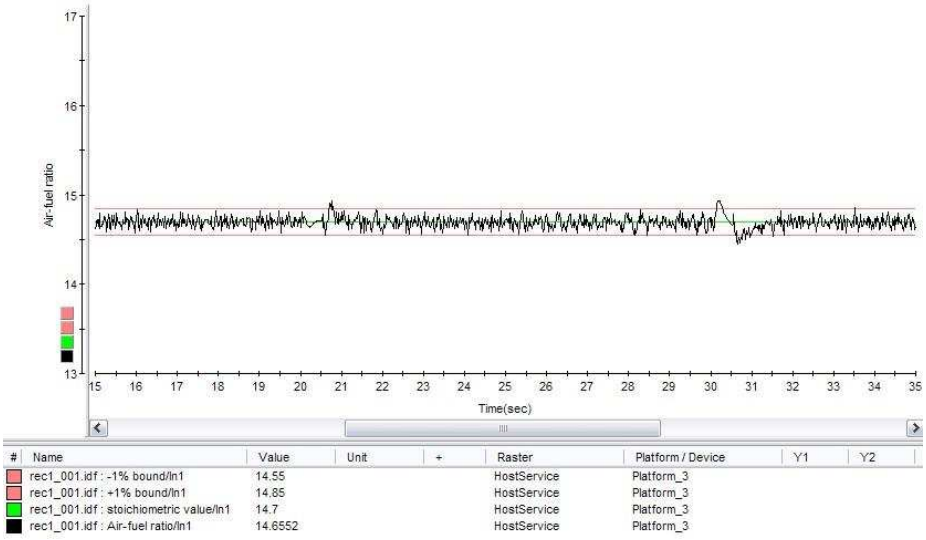

Figure 8(a) Air-fuel ratio control result of the NMPC using the modified Volterra model

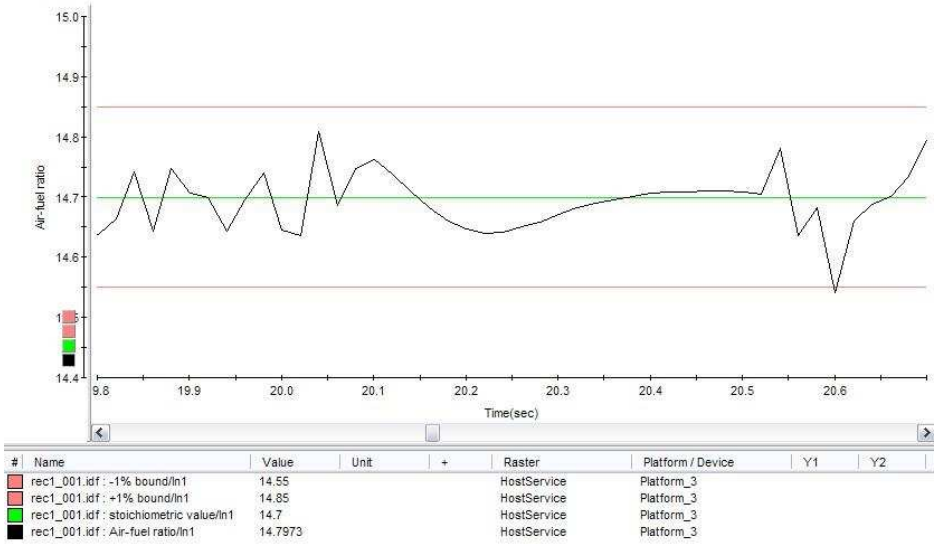

Figure 8(b) Augmented image of Figure 8(a) at t=20sec

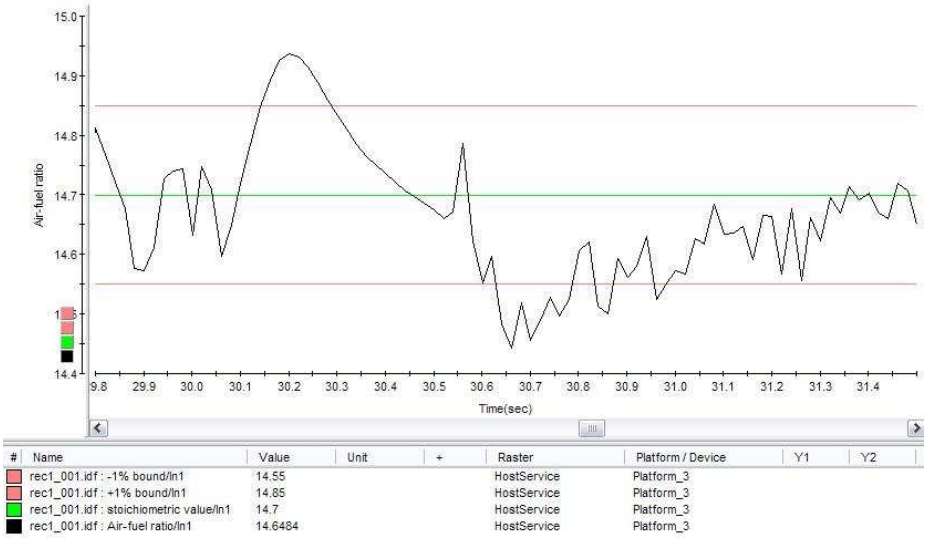

Figure 8(c) Augmented image of Figure 8(a) at t=30sec

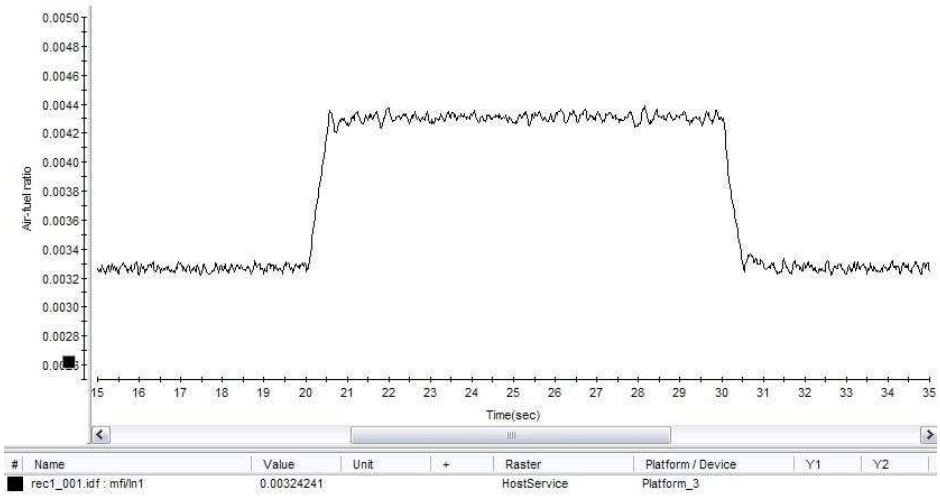

Figure 9 The injected fuel mass flow rate produced by the NMPC

In order to demonstrate the effectiveness of the proposed modified Volterra model-based NMPC, its performance is compared with that of a feed-forward plus feedback control. The feed-forward controller is implemented by a look-up table while the feedback control is implemented by a PI controller. The look-up table uses the throttle angle only as the input and the gains in the table are obtained from the MVEM Simulink model in the steady state. The digital PI controller of the form in (51) [14] is used.

$$\dot{m}_{fi}(k) = \dot{m}_{fi}(k-1) + K_s\left[\left(1+\frac{T_s}{\tau_I}\right)e(k) - e(k-1)\right] \tag{51}$$

The sampling time T_s is 0.02s, which is the same as that in the NMPC. The feedback signal of the

system is the measured AFR value with time delay. Open-loop Ziegler-Nichols method is used to set initial PI controller parameters. After the fine tuning of the PI controller parameters, the performance and corresponding injected fuel mass flow rate are shown in Fig. 10 and Fig. 11 respectively. The tuned controller parameters are the gain $K_s = 3.25 \times 10^{-4}$ and the integral time $\tau_I = 0.22$ sec .

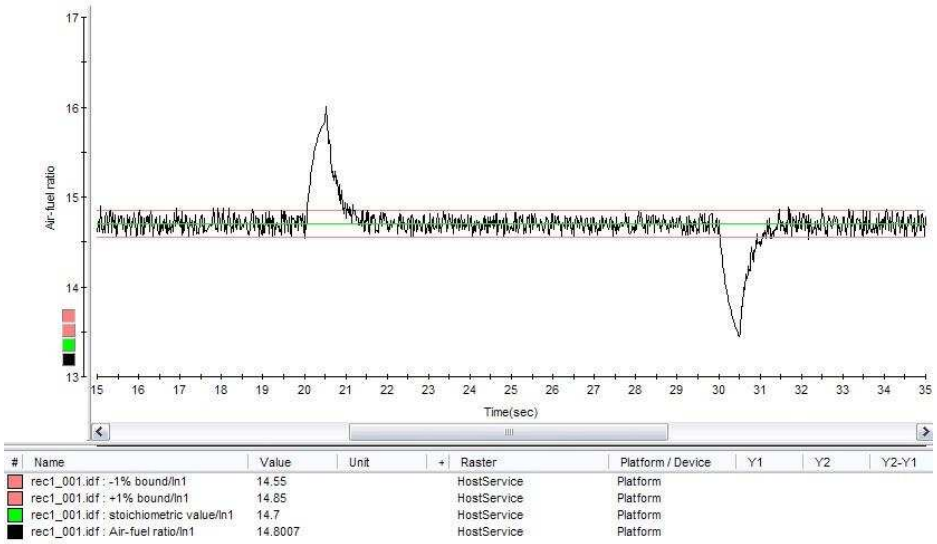

Figure 10 Air-fuel ratio control result of the feed-forward plus feedback control

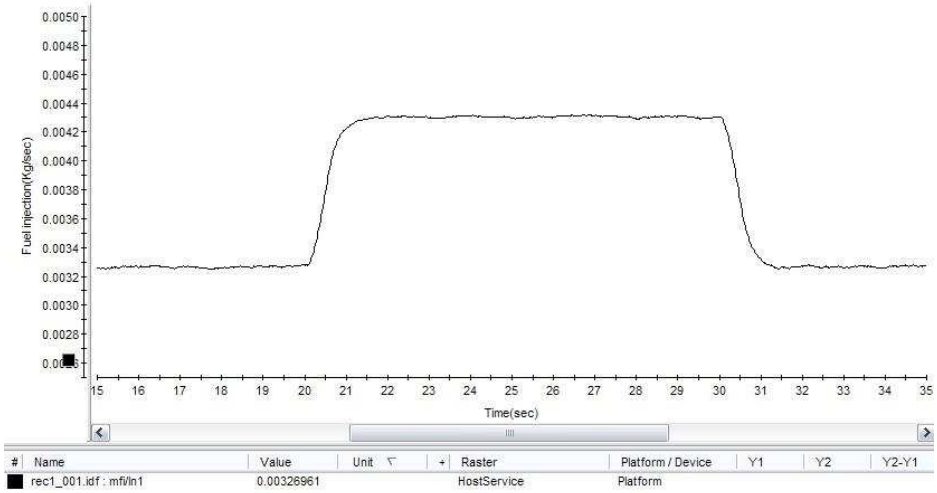

Figure11 The injected fuel mass flow rate produced by the feed-forward plus feedback control

The steady state response of the feed-forward plus feedback control is acceptable, as it converges to the stoichiometric value of 14.7. However, the performance in transient state is not as good as the NMPC. It exhibits large overshoot and relatively longer settling time. The step change of throttle angle at t=20sec caused the AFR value to reach 16.1, which is well exceed the required 1% boundary. After that, the AFR recovers slowly and it takes approximately 1.3 seconds before returning to the required region $14.7(1 \pm 1\%)$. At time instant of 30 seconds, the step change of the

throttle angle caused the AFR value dropping to 13.4, which is out of required 1% boundary. The settling time for the AFR to settle back into $14.7(1\pm1\%)$ region takes about 1.1 seconds.

Comparatively, the NMPC control has achieved much better performance than the feed-forward plus feedback control in terms of the transient dynamics. From Fig. 8 and Fig. 9 above, it can be seen that the maximum overshoot at t=20s and the maximum undershoot at t=30s caused by the change in throttle angle are much smaller than the feed-forward plus feedback control. As shown in Fig. 8(b) the maximum overshoot is 14.75 and the maximum undershoot is 14.63. The settling time of the response is 0.4 seconds and during all this time the AFR remains within the $14.7(1\pm1\%)$ region. And in Fig.8(c), on the t=30s onwards, the maximum overshoot is 14.95 and minimum undershoot is at 14.42. It took around 1.1 seconds for the AFR to settle back within the $14.7(1\pm1\%)$ region.

6 Conclusions

This paper investigates the application of the Volterra model to IC engine modelling and nonlinear model predictive control. A modified Volterra model is developed in this work to cope with the significant response speed difference to different engine inputs. With the modified Volterra model, the modelling error is kept the minimum while the truncation order is greatly reduced. With the modified Volterra model, the model prediction formulas in the NMPC algorithm are developed. The optimization in the MPC is solved using the active set method. The Volterra model is adapted on-line with the recursive Least Squares algorithm to model the time varying dynamics or post-fault dynamics caused by any malfunction or mechanical wear of engine components. This is very important to avoid the control performance to degrade due to aging and environment change. By comparing with the feed-forward plus feedback control, it is shown that the NMPC based on the modified Volterra model performed much better than the feed-forward plus feedback control in transient response. Real-time simulations prove that the NMPC algorithm can be executed in real time within one sampling period. It proves that the modified Volterra model based NMPC is a promising control algorithm with potential to replace the feed-forward plus feedback control of the production ECU to control air-fuel ratio in IC engines.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under the Grant 51075175 and the Research Fund from State Key Laboratory of Automobile Dynamic Simulation.

References

Balluchi A, Benvenuti L, Benedetto D, Pinello MD, Sangiovanni C, Vincentelli AL (2000) Automotive engine control and hybrid systems: challenges and opportunities. *Proceedings of the IEEE*, 88(7): 888-912. (1)

Bryon R, Maner FJ, Babatunde DI, Ogunnaike A, Pearson RK (1996) Nonlinear model predictive control of a simulated multivariable polymerization reactor using second-order Volterra models. *Automatica*, 32(9): 1285-1301. (11)

Butt Q R, Bhatti A L, Mufti M R, Rizvi M A, Awan I (2013) Modeling and online parameter estimation of intake manifold in gasoline engines using sliding mode observer, *Simulation Modelling Practice and Theory*, Volume 32, March 2013, Pages 138-154

Chen H, Gong X, Hu Y F, Liu Q F, Gao B Z, Guo H Y (2013) Automotive Control: the State of the Art and Perspective, *Acta Automatica Sinica*, Volume 39, Issue 4, April 2013, Pages 322-346

Choi SB, Hendrick JK (1998) An observer-based controller design method for improving air/fuel characteristics of spark ignition engines. *IEEE Trans. on Control Systems Technology*, 6(3): 325-334. (7)

Coen T, Anthonis J and De Baerdemaeker J (2008) Cruise control using model predictive control with constraints. *Computers and Electronics in Agriculture*, 63: 227-236. (18)

Doyle FJ, Pearson RK, and Ogunnaike BA (2001) Identification and control using Volterra models. *London: Springer*. (16)

Fletcher RR (1987) Practical methods of optimization, 2nd edition. *Wiley*. (19)

Gruber JK, Oliva A (2012) Nonlinear MPC for the airflow in a PEM fuel cell using a Volterra series model. *Control Engineering Practice*, 20: 205-217. (12)

Gruber JK, Rodríguez F, Bordons C, Berenguel M, Sánchez JA (2011) Nonlinear MPC based on a Volterra series model for greenhouse temperature control using natural ventilation. *Control Engineering Practice*, 19: 354-366. (13)

Hendricks E (2000) A generic mean value engine model for spark ignition engines. *Proc. of 41st Simulation Conference (*SIMS 2000), DTU, Lyngby, Denmark. (15)

Hendricks E, Chevalier A, Jensen A, Sorenson SC (1996) Modelling of the intake manifold filling dynamics. *SAE paper 960037*, 14: 236-245.

Laguerre–Volterra observer. *Trans. of the Institute of Measurement and Control*, 31(2): 129–151.

Ljung L (1999) System Identification-Theory for the User, second ed. *Prentice-Hall, Englewood cliffs(NJ)*: p. 361-369. (17)

Manzie C, Palaniswami M, Watson H (2001) Gaussian networks for fuel injection control. *Proc. I. Mech. E. Part D: Journal of Automobile Engineering*, 215(10): 1053-1068. (5)

Manzie C, Palaniswami M, Ralph D, Watson H, Yi X (2002) Model predictive control of a fuel injection system with a radial basis function network observer. *Trans. of ASME: J. of Dynamic Systems Measurement and Control*, 124(4): 648-658. (6)

Nicolao D, Scattolini G, Siviero R (1996) Modelling the volumetric efficiency of IC engines: parametric, non-parametric and neural techniques. *Control Engineering Practice*, 4(10): 1405-1415. (2)

Shah MZ, Samar R and Bhatti AI (2015) Lateral track control of UAVs using the sliding mode approach: from design to flight testing. *Trans. of the Institute of Measurement and Control*, 37(4): *457-474.*

Tan Y, Mehrdad S (2000) Neural-networks-based nonlinear dynamic modeling for automotive engines. *Neurocomputing*, 30: 129-142. (3)

Vinsonneau JAF, Shields DN, King PJ, Burnham KJ (2003) Polynomial and neural network spark ignition engine intake manifold modeling. Proc. of the 16th Int. Conference on Systems Engineering, Coventry, U.K., pp. 718-732. (4)

Wang SW, Yu DL, Gomm JB, Page GF and Douglas SS (2006a) Adaptive neural network model based predictive control for air-fuel ratio ofICengines. *Engineering Application of Artificial Intelligence*, 19: 189-200. (9)

Wang SW, Yu DL, Gomm JB, Page GF and Douglas SS (2006b) Adaptive neural network model based predictive control of an internal combustion engine with a new optimization algorithm. *Proc. I. Mech. E. Part D: J. of Automobile Engineering*, 220(2): 195-208. (10)

Yoon R, Sunwoo M (2001) An adaptive sliding mode controller for airfuel ratio control of spark ignition engines. *Proc. I. Mech. E. Part D: J. of Automobile Engineering*, 215: 305-315. (8)

Zhang HT, Tischenko L, Yu PZ (2009) A novel adaptive control algorithm based on non-linear

Appendix

Notation		P_f	friction power (kW)
		P_i	manifold pressure (bar)
H_u	fuel lower heating value (kJ / kg)	P_p	pumping power (kW)
I	crank shaft load inertia (kJm^2)	R	gas constant (here 287×10^{-5})
L_{th}	stoichiometeric air/fuel ratio (14.7)	T_a	ambient temperature (K)
$\dot{m}_{ap}$	air mass flow rate into intake port (kg / s)	T_{EGR}	EGR temperature (K)

$\dot{m}_{at}$	air mass flow rate past throttle plate (kg / s)	T_i	intake manifold temperature (K)
$\dot{m}_{EGR}$	EGR mass flow rate (kg / s)	t_d	time delay of fuel injection systems
$\dot{m}_f$	engine port fuel mass flow rate (kg / s)	V	throttle position (degrees)
$\dot{m}_{ff}$	fuel film mass flow rate (kg / s)	V_d	engine displacement
$\dot{m}_{fi}$	injected fuel mass flow rate (kg / s)	V_i	Manifold + port passage volume (m^3)
$\dot{m}_{fv}$	fuel vapour mass flow (kg / s)	η_i	indicated efficiency
n	crankshaft speed (krpm)	λ	air/fuel ratio
P_a	ambient pressure (bar)	$\Delta\tau_d$	injection torque delay time (s)
P_b	load power (kW)	κ	Ratio of the specific heats=1.4 for air